Vol.2015-CE-128 No.7
2015/2/14

学生用PCを用いたユーザPCコンピューティングシステムのWebインターフェース

金奉洙[1,a)]　舩曳 信生[1,b)]

概要：本グループでは，研究室内の学生が使用するパーソナルコンピュータ（PC）の空き資源を利用して，大規模シミュレーションなどの計算プロジェクトを実行する，**ユーザ PC コンピューティングシステム UPC**（User-PC Computing System）を開発している．UPC では，大規模コンピューティング環境の実現に関して，運営コストの低減と利便性の向上の実現を目指している．UPC では，計算プロジェクトの実行には，コマンド入力が必要となっており，その知識に乏しいユーザには UPC の使用が容易ではない．そこで本研究では，ブラウザで UPC の利用を可能とするために，Web 技術を用いた UPC ユーザインターフェース機能を開発する．UPC サーバ間とファイル共有ソフトを用いて連携することで，異なる複数の UPC へのインターフェース機能の提供を可能としている．本稿では，今回開発した UPC インターフェース機能の構成を示す．

キーワード：グリッドコンピューティング，デスクトップグリッド，Web，ユーザインターフェース，ユーザ PC

1. まえがき

グリッドコンピューティング（以下，グリッドと略）では，ネットワークで接続されている複数のコンピュータを連携して一つの仮想スーパーコンピュータを実現する．グリッドでは，利用する資源の種類や方法により，以下のように分類される．実際のプロジェクトでは，これらが混用されている．

- 計算グリッド
  地域的に分散されているコンピューティング資源（CPU，メモリ）を利用して大規模な問題を解析するためのシステムで，1 秒間に数億，数十億個の作業を同時に行うことが出来る．
- データグリッド
  大容量のデータを，ネットワークを通じて共有するために仮想的なストレージ環境を構築する．
- アクセスグリッド
  地理的に離れているユーザ間で，オーディオやビデオを利用して，仮想的に協業環境を提供する．

近年のパーソナルコンピュータ（PC）の性能向上により，グリッドの一種として，インターネットなどのネットワークを介して，複数の PC の遊休計算資源を結びつけ，仮想的に一つの複合したコンピュータシステムとしてサービスを提供する，ボランティアコンピューティング（VC）が注目されている．VC で提供されるサービスには，大規模な演算が必要なシミュレーションやたんぱく質の分析，暗号解読などが含まれている [1]．

VC では，システムの計算資源を利用する**ユーザ**（クライアント），グリッドの各機能を提供する**ミドルウェア**，計算資源を提供する**ワーカ**の 3 つの要素で構成される．ユーザは，要求する計算プロジェクトをグリッドに登録し，ワーカからの計算結果を得る．ミドルウェアは，アプリケーションプログラムとオペレーティングシステムの中間に位置するソフトウェアであり，各ワーカの仕様の違いを吸収し，それらを統一的に利用するためのプログラムである．代表的なミドルウェアには，事実上のスタンダードで知られている Globus ツールキットがある [2]．ワーカは，実際に計算に使われる資源を提供するコンピュータであり，ネットワークに接続された PC などが該当する．

しかしながら，VC では，インターネットを通じてボランティア的に提供される PC を利用するため，計算結果に信頼性が低いといった問題がある．その原因としては，ワーカに使用される PC のウィルス感染やハードウェア故障に加え，VC の妨害者からの意図的に誤った計算結果が返さ

[1] 岡山大学大学院自然科学研究科
〒 700–8530 岡山市津島中 3–1–1
[a)] pusw85zy@s.okayama-u.ac.jp
[b)] funabiki@okayama-u.ac.jp

れることなどが挙げられる．

VC の問題点の対策として，本グループでは，**ユーザ PC コンピューティングシステム UPC**（User-PC Computing System）を提案している．UPC は，VC において，計算資源を提供するワーカを同一組織内の PC，例えば研究室内の PC などに限定したシステムである．UPC では，信頼できるワーカの資源のみを利用することで，VC の低い信頼性の改善を図っている．

現在 UPC は，Linux を対象としたシステムの実装を終え，性能評価を進めている．その結果，サーバにリクエストが集中した場合にも実用的な時間で処理が完了すること，ワーカでユーザが指定した CPU 使用率で計算が実行できることを確認している．後者では，実際のワーカの利用状況を想定し，ユーザによる作業中の PC においても正常にプロジェクトの計算が処理されることを確認している [3]．しかしながら，計算プロジェクトの実行には，コマンド入力が必要であり，その知識に乏しいユーザには UPC の使用が容易ではないといった問題がある．

そこで本研究では，Web ブラウザを用いて UPC の利用を可能とするために，Web 技術を用いた UPC ユーザインターフェース機能を開発する．インターフェース用 Web サーバと UPC サーバ間を，ファイル共有ソフトを用いて連携することで，複数の UPC サーバへのインターフェース機能の提供を可能としている．本稿では，今回開発した UPC インターフェース機能の構成を示す．

ここで，ユーザインターフェースは，人と機械やプログラムの間でコミュニケーションが出来るようにするための手段を表す．一般に，あるプログラムを利用するユーザは，そのプログラムの内部構造や仕組みには興味を持たない．インターフェースで提供される情報で，そのプログラムを受け入れて使用する．そのため，使いやすく，分かりやすいインターフェースを提供することがシステムに重要である．

以下，2 章では，UPC の概要を示す．3 章では，本研究で用いた要素技術を紹介する．4 章では，本研究で提案する UPC の Web インターフェースについて述べる．最後に，5 章で本研究のまとめと今後の課題を述べる．

2. UPC の概要

本章では，本グループで提案・実装している UPC の基本モデル．基本機能，構成要素について述べる [3]．

2.1 基本モデル

本研究では，UPC の基本モデルとして，**図 1** に示すマスタ・ワーカモデルを採用している．マスタには UPC の管理サーバ（UPC サーバ），ワーカには遊休資源を提供する研究室メンバーの PC（ワーカ PC）としている．マスタは，計算プロジェクトと呼ばれる全体の計算の進行を管理する．ワーカは，マスタから割り当てられた計算を行う．その際，複数のワーカに計算ジョブを割り当てることで，並列計算を実現している．

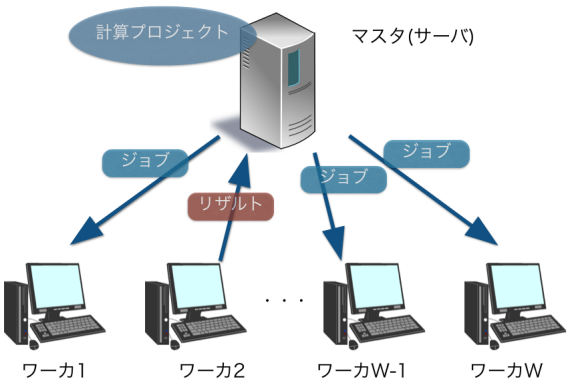

図 1 マスタ・ワーカモデル
Fig. 1 Master-worker model.

2.2 基本機能

UPC の基本機能の流れを示す．

(1) 計算プロジェクトのエントリー
UPC のユーザは，実行したい計算プロジェクトの**ソースコードファイル**，コンパイルを含むその実行コマンドを記述した**スクリプトファイル**，実行時のパラメータ値を記述した**パラメータファイル**を準備し，UPC サーバに登録する．その際，複数のパラメータ値を実行したい場合，その値毎に異なるパラメータファイルを作成し，登録する．

(2) ジョブのスケジューリング
計算プロジェクトのファイルの登録後，ユーザは，UPC サーバに計算実行コマンドを入力する．UPC サーバでは，ソースファイル，スクリプトファイルと，1 つのパラメータファイルを，**ジョブ**と呼ばれる単位に分割する．その後，現在稼動しているワーカ PC の情報を基に，ジョブの実行先を決定する．この機能は，ジョブ管理システム（JMS）と呼ばれている．

(3) ジョブの配信
UPC マスタは，JMS で決定したジョブとその実行先を読み込み，ジョブをそのワーカ PC に送信する．

(4) ジョブの実行
ジョブを受け取ったワーカ PC は，スクリプトファイルを実行することで，ソースファイルのコンパイル，実行を行う．この時，ワーカ PC では，その PC の所有者が指定した CPU 使用率，メモリ使用量の制限下でジョブを実行する．

(5) リザルトの回収
ジョブの実行を終えたワーカ PC は，自動的に計算結果（リザルト）を UPC サーバに返却する．

(6) リザルトの集計
UPC サーバでは，返却されたリザルトの集計を行う．ユーザは，集計結果を閲覧することにより要求した計算プロジェクトの進行状況を確認する．

2.3 サーバ機能

UPC サーバの機能を示す．

(1) マルチスレッド
UPC サーバでは，計算プロジェクトの受け付け処理 (request)，ジョブ管理 (job manage)，ジョブ送信処理 (job send), リザルト受信処理 (result recv) が，それぞれ，個別のスレッドで実装されている．スレッド間データのやり取りにはキューを使いることで，最初に投入されたデータから取り出す FIFO(First In First Out) アルゴリズムを採用している．各キューにおいて，データが入出力される際には，ロックをかけることで，競合の発生を防いでいる．以下に，各スレッドの処理を示す．

- 受付処理スレッド
  ワーカ PC からの通信リクエストを常に受け付け，そのリクエストに対応する処理スレッドのキューに格納する．
- ジョブ管理スレッド
  計算プロジェクトの実行を行うために，ジョブ管理システムを呼び出す．
- ジョブ送信処理スレッド
  ジョブ管理システムから受け取った腕先の IP 情報とジョブ情報を基に，ジョブを送信する．
- リザルト受信スレッド
  キューからリザルトの返却情報を読み込み，ワーカからリザルトを受信する．

(2) ワーカ PC 情報管理
PC が，その起動などにより新たに UPC に参加した場合，そのワーカ PC の情報を UPC サーバに送信する．ワーカ PC の情報には，合計メモリ量，使用可能メモリ量，CPU 周波数・コア数，ディスク容量が含まれている．また，UPC サーバがあるワーカ PC と通信できなくなったとき，その情報が削除される．

(3) ジョブ管理システム（JMS）
JMS では，ジョブ管理スレッドからの実行コマンドの読み込み，実行コマンドに対するワーカ PC の割当を行う．後者では，処理待ちのジョブリスト（キュー）からジョブを 1 つ取り出し，起動中でジョブを実行してないワーカ PC の中で，CPU 周波数の最も高いものに割り当てる．

2.4 ワーカ機能

UPC のワーカ PC の機能を示す．

(1) マルチスレッド
ワーカ PC では，起動処理 (main), ジョブ受付処理 (request), ジョブ受信処理 (job recv), ジョブ実行処理 (job execute)，計算結果送信処理 (result send) の各処理が，それぞれ，個別のスレッドで実装されている．UPC サーバと同様，スレッド間のデータのやり取りにはキューを利用している．各スレッドでの処理を以下に示す．

- 起動処理スレッド
  ワーカ PC で UPC のジョブを起動したとき，その PC の情報を UPC サーバに送信する．また，定期的に PC のリソース使用状況をサーバに送信する．
- 受付処理スレッド
  UPC サーバからの計算リクエストを常に受付ける．新しいリクエストを受け付けた場合，それに対応する処理スレッドのキューに格納する．
- ジョブ受信処理スレッド
  受付処理スレッドからの計算リクエストの受け取る時に，そのジョブの情報をジョブ実行処理スレッドのキューに格納する．
- ジョブ実行処理スレッド
  実行処理スレッドのキューの先頭のジョブ情報を読み込み，ジョブを子プロセスとして実行する．ジョブの実行終了時に，そのジョブの情報をリザルト送信スレッドのキューに格納する．
- リザルト送信処理スレッド
  リザルト送信スレッドのキューから，実行終了ジョブの情報と生成されたリザルトの情報を読み込み，それらを UPC サーバに送信する．

(2) リソース制御
ワーカ PC では，UPC のジョブを実行する際に使用する PC のリソースを制限するために，cgroup-lite パッケージを使用する．このパッケージでは，Ubuntu で提供されるもので，CPU 時間，システムメモリといった PC の資源を，プロセスグループ単位で割り当てることを可能としている．また，cgroup-lite は，階層的に構成されており，子 cgroup が親 cgroup の属性の一部を継承することが可能となっている．これにより，UPC ワーカ (親 cgroup) が生成したジョブのプロセス (子 cgroup) に，リソースの割当上限を継承させることが可能となる．

(3) デーモン機能
ワーカ PC では，また，Upstart と呼ばれる UNIX 系 OS のデーモン機能を利用している．この機能により，PC が起動した際の，UPC のワーカプロセスの自動起動を可能としている．

3. 利用した要素技術

本章では，Web を用いた UPC のユーザインターフェース機能の実装に利用した要素技術を紹介する．

3.1 Bittorrent Sync によるサーバ間連携

まず，従来の UPC サーバと，今回開発するインターフェース用 Web サーバの連携のために採用した Bittorrent Sync について述べる．

3.1.1 Bittorrent

Bittorrent Sync は，P2P 方式のファイル共有プロトコルとして，現在最も使用されている，*Bittorrent* を採用している．Bittorrent では，ファイルをダウンロードする際に，1 つのサーバ（ピア）からだけではなく，そのファイルを有する複数のピアからダウンロードする．また，Bittorrent では，ダウンロード先のピアとの通信に，再送を含めての時間が掛かりすぎることを防ぐために，ファイルを 1/4 メガバイトサイズのピースに分割して送信する．このファイル分割により，複数ピアからの同時ダウンロードを可能としている [4], [5].

3.1.2 Bittorrent Sync

Bittorrent Sync は，Bittorrent プロトコルを開発した Bittorrent 社から公開されている，フォルダ間のファイル同期用のフリーソフトウェアである．Bittorrent プロトコルに基づいて，複数のピアの指定フォルダ間で，そこに保存されているファイルの同期（最新データへの更新）を行う．その際，同期するファイルの大きさや数には制限が無い．また，Windows，Linux，Mac など，様々な OS で利用することができる．特に Linux では，Web ブラウザで利用するための API も提供されている [6].

Bittorrent Sync では，**シークレットキー**を用いて，ファイルを同期するピアのフォルダをお互い認識し合う．シークレットキーは，コマンド/dev/random（Mac，Linux），Crypto API（Windows）を使用してランダムに生成される 20 バイトの長さの文字列である．十分な長さを有するため，重複したシークレットキーが生成される可能性はほぼゼロである．このシークレットキーには，読み取り専用と読み取り/書き込み用の 2 つが存在し，ファイル同期の用途・目的によって，これらを使い分ける．

Bittorrent Sync を用いてファイルを同期するには，同じシークレットキーを有するピアを見つける必要がある．その手段を以下に示す．これらの手段は，Bittorrent Sync での設定により，有効または無効とすることができる．

- ローカルピアの探索
  ローカルネットワーク内の全てのピアは，ブロードキャストパケットの送信によって探索される．同じシークレットキーを持っているピアがそのパケットに応答することでそれらが接続される．
- ピアエクスチェンジ（PEX）によるピア情報交換
  2 つのピアが接続されると，各自で把握しているピアに関する情報を交換する．
- 既知のピアとの接続
  静的な IP アドレスとポート番号を持つピアを知っている場合には，その情報を直接入力することで接続できる．
- トラッカーサーバ (Tracker Server) の利用
  トラッカーサーバとは，ピア間に位置し，ピア同士を接続させるためのサーバである．このサーバは NAT トラバーサル（Network Address Translation traversal ）の実行を支援するため，NAT の下に位置するピアとも直接接続することができる．

同期されるファイルは，遠隔のサーバには保存されず，同期を行うピア（デバイス）にのみに保存される．また，同期時のファイル転送は，AES 暗号方式によって暗号化されている．

3.2 システムの実装環境

本研究の Web サーバは，Vmware 上の仮想 OS として Ubuntu12.04 をインストールし，その上で Apache と Tomcat6 を利用した．仮想環境上で開発を行なうために，軽量て拡張性の高い IDE として Sublime Text2 を用いた．開発言語として，サーバサイドでは Servlet と JSP，クライアントサイドでは HTML と javaScript のライブラリである jQuery を主に利用した．データベースとして PostgreSQL9.1.14 を用いた．

3.3 Ajax

Ajax とは，Asynchronous JavaScript + xml の略で，Web サーバとブラウザ間の非同期通信を利用して，データの取得や動的な Web ページの書き換えを行う技術である．Ajax は，2005 年に Jesse James Garrett 氏によって初めて名付けられ，Google が提供した地図サービスが Ajax を用いてスムーズに情報を表示させたことからブームが始まった．

従来の Web アプリケーションでは，ユーザの操作により Web サーバにリクエストを送ってから新しいページを取得してページ全体を書き換えるため，ページの取得中には何も操作できなくなるといった問題があった．しかし，Ajax を利用することで，バックグラウンドでサーバと非同期通信を行うため，この問題を解消すると共に，Web ページ中の必要な箇所のみの書き換えで済み，高速での更新が可能となっている．Ajax を用いて Web サーバとブラウザ間でデータをやり取りする際に，以前は XML 形式のみが利用されたが，最近は JSON フォーマットが最も使用されるようになっている [7].

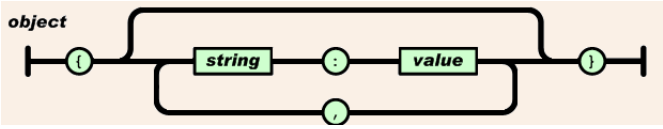

図 2　JSON オブジェクトの形式
Fig. 2　Format of JSON object.

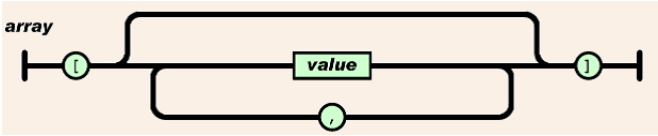

図 3　JSON 配列の形式
Fig. 3　Format of JSON array.

3.4　JSON

通常，プログラミング言語毎にデータの記述形式が異なっており，異なる言語間には互換性がない．異なる言語間でデータの受け渡しを行うために，XML が利用されている．しかし，XML では文法が複雑で特定のパーザを必要とする上，厳しい表現規則によってデータのサイズが大きくなるといった問題がある．そこで提案された形式が JSON である．JSON は JavaScript Object Notation の略で，テキストベースのデータフォーマットである．多くのプログラミング言語において，JSON は単純な処理で書き出し，読み込みができるため，異なる言語間でのデータの受け渡しに利用されている．特に，Web アプリケーションで盛んに利用されている [8].

JSON では，オブジェクト (object)，配列 (array)，値 (value)，文字列 (string)，数値 (number) と呼ばれる多様な形式を持っている．その中で，今回の実装で使用したオブジェクトと配列の形式について述べる．

オブジェクトは，順序付けされない名前／値のペアのセットである．“{”(左の中括弧) で始まり，“}”(右の中括弧) で終わる．各名前の後ろには，“:”(コロン) が与えられ，名前／値のペアは，“,”(カンマ) で区切られる（**図 2** 参照）．

配列は，順序付けされた値の集合である．“ [”(左の大括弧) で始まり，“]”(右の大括弧) で終わる．値は，(カンマ) で区切られる (**図 3** 参照).

3.5　Bootstrap3

Web アプリケーションのユーザインターフェース（UI）の開発は，実際には容易ではない．ユーザの利用する端末として単一の環境のみを考慮する他の UI と違い，Web の UI は，様々な Web ブラウザの環境において互換性を満たす必要がある．いわゆる，クロスブラウジング (Cross Browsing) 作業が必要となる．

本研究では，その対策として，CSS のフレームワークとして最も使用されている Bootstrap を利用した．Bootstrap は，ツイッター (Twitter) でオープンソースとして公開された，Web のフロントエンド開発ツールである．約 7000 行で書かれており，Web ページを作るためのほとんどの要素が定義されているため，HTML ページのタグに適用するだけで，Bootstrap で指定したデザインに Web ページが自動的に表示される．さらに，スマートフォンなど，端末の画面サイズに併せて自動的にデザインを変更する，レスポンシブ Web デザイン機能も実装されているため，同一画面をサイズ毎に作成する必要がない [9].

3.6　Apache Commons Fileupload

Web インターフェースを用いて，UPC に計算プロジェクトを依頼するには，プロジェクトのソースファイルやパラメータファイルを Web サーバにアップロードする必要がある．そのために，今回，オープンソースの Apache Commons FileUpload ライブラリを使用した．このライブラリは，Apache のプロジェクトの一種である Apache Commons で公開されており，Web アプリケーション実装時にファイルのアップロード処理を簡単にするために開発されている．本機能の実装には，クライアントサイドでは，HTML ファイルにおいて POST メソッドおよび Content Type を multipart/form-data で指定して HTTP リクエストを転送するだけでよい．一方，サーバサイドでは，HTTP リクエストをパース (Parse) すると，クライアントから転送されてきたファイルやテキストを，既に実装されている FileItem インターフェースを用いて簡単に処理することができる [10].

4.　ユーザインターフェースの提案

現在の UPC は，Linux 環境で開発されており，UPC の利用者は分散処理を含む，Linux に関するある程度の知識が必要である．そこで本研究では， UPC の利用を容易とするために，UPC に対する Web ユーザインターフェースの提案を行う． Web ブラウザを利用することで，簡単に計算プロジェクトの依頼が可能となる．

4.1　提案システムの構成

まず，提案システムの構成について述べる．本システムは，ユーザのホスト（Web ブラウザ)， ユーザインターフェース用 Web サーバ，UPC サーバ，ワーカ PC で構成されている（**図 4** 参照）.

UPC のユーザは，Web ブラウザを用いて，インターネット経由で Web サーバにアクセスし，依頼したい計算プロジェクトをアップロードする．Web サーバは， Bittorrent Sync を用いた UPC サーバとのファイル同期により，計算プロジェクトの UPC サーバへの送信，そのリザルトの受信を行う． Web サーバおよび UPC サーバでは，計算プロジェクトの各ファイルの同期のために，同一のディレクトリ構成を構築している．その構造を**図 5** に示す.

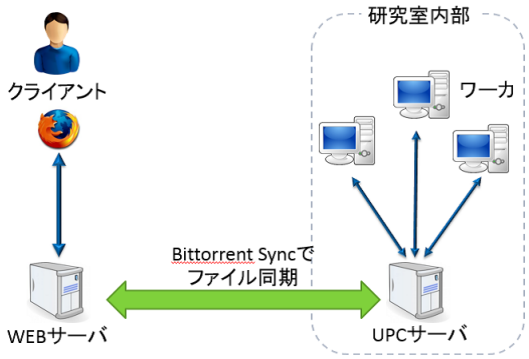

図 4　提案システムの構成
Fig. 4　Outline of proposed system.

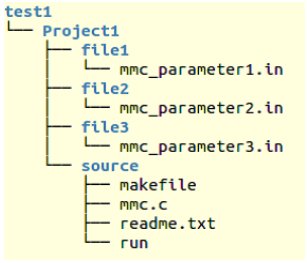

図 5　ファイル同期のためのディレクトリ構造
Fig. 5　Directory architecture for file synchronization.

ここでは，ユーザが本システムへのログイン時に入力した ID 名のフォルダ（図 5 では test1）が最上位に配置される．その下位には，アップロードする際に入力する計算プロジェクト名のフォルダ（Project1），その下に一つのソースファイル用フォルダ（source）と複数のパラメータ用のフォルダ（file*）が配置される．そして，ソースファイルやパラメータファイルは，それぞれのフォルダの下に配置される．これにより，ファイル同期による，Web サーバ・UPC サーバ間での計算プロジェクトの受け渡しを実現している．

Web サーバでは，ユーザからアップロードされたファイルを保存するフォルダとは別に，UPC サーバと同期するためのフォルダを設けている．これは，ユーザが Web サーバにファイルをアップロードした後に，UPC サーバに送信する前に，それらの内容を確認できるようにするためである（**図 6** 参照）．

4.2　インターフェース画面設計

本システムの画面遷移を**図 7** に示す．画面の上段に常時，メニューが表示され，これにより，「ステート」，「アップロード」，「実行」の画面に遷移できる．本節では，それぞれの画面における操作の詳細について述べる．

4.2.1　ログインと会員登録画面

ログイン画面，会員登録画面を，**図 8**，**図 9** に示す．ログイン画面では， ID，パスワードを入力することでログインを行なう．画面下段に配置されている「Register」ボタンを押すと，会員登録画面に遷移する．会員登録画面では，ID，パスワード，パスワード確認欄が表示される．「id check」ボタンを押し，システムに登録済みの ID との重複を確認した後，会員登録を行う．その後，パスワードを 2 回入力し，それらが一致するまで，パスワードの入力を繰り返す．

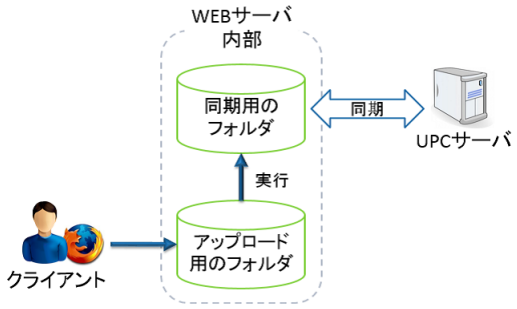

図 6　Web サーバ内の 2 種類のフォルダ
Fig. 6　Two-types folders in Web server.

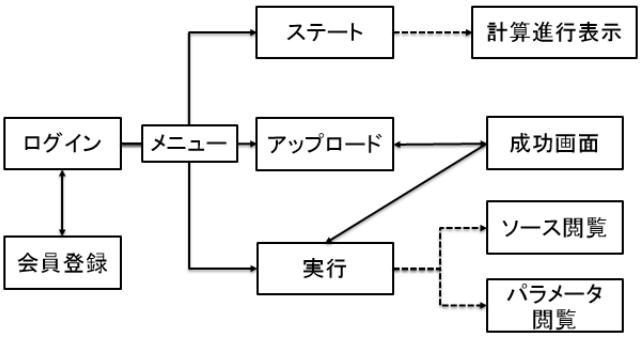

図 7　提案システムの画面遷移
Fig. 7　Screen transitions in proposed system.

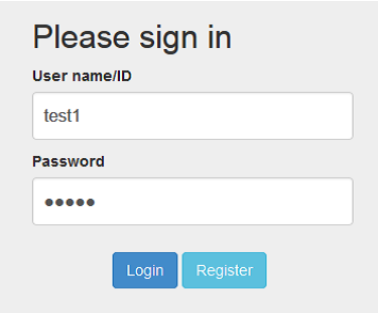

図 8　ログイン画面
Fig. 8　Login screen.

4.2.2　アップロード画面

アップロード画面を，**図 10** に示す．アップロード画面では，計算プロジェクトのソース，パラメータ，スクリプトの各ファイルを，Web サーバにアップロードする．それに加えて，プロジェクトの名称，コメント，各パラメータファイルに対するコメントなどの入力を行う．

計算プロジェクトを UPC で実行するためには，実行したいプログラムのソースファイル，そのコンパイル，実行

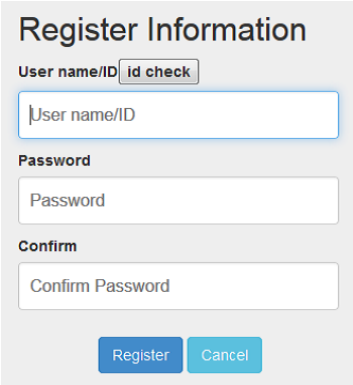

図 9　会員登録画面
Fig. 9　Register screen.

図 10　アップロード画面
Fig. 10　Upload screen.

コマンドを記述した makefile，run の名称を有するスクリプトファイルが含まれなければならない．そのため，選択したソースファイル中にこれらすべてのファイルが揃っていない場合，エラーを表示し，アップロードを実行しない．その際，ソースファイルが圧縮されている場合，アップロード時に解凍した後，すべてのファイルが揃っていることを確認する．

アップロードを成功すると，Web サーバ上に図 5 のディレクトリ構造で保存される．ここで，この段階では，これらのファイルは Web サーバに保存されるのみであり，UPC サーバには送られていない．

4.2.3　実行画面

実行画面を，**図 11** に示す．本画面では，Web サーバにアップロードされた計算プロジェクトのリストを表示する．その 1 行目に，計算プロジェクトの名称とコメント，2 行目からは，各パラメータファイルの情報として，そのパラメータファイルのフォルダ名，コメント，アップロード時刻が表示される．

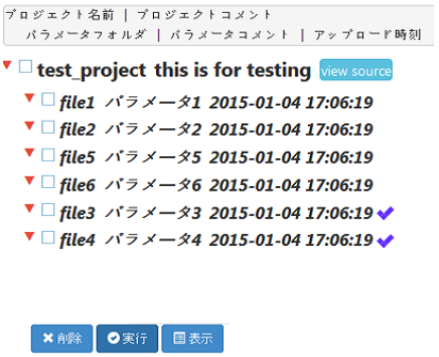

図 11　実行画面
Fig. 11　Execution screen.

図 12　ソースファイル閲覧ダイアログ
Fig. 12　Source file browsing dialog.

実行画面では，アップロードされているファイルの名称や内容を閲覧できる機能を，ダイアログ形式で提供している．計算プロジェクトのコメント欄の隣に配置している「view source」ボタンでソース確認のダイアログ (**図 12**) が表示される．また，画面の下に配置している「表示」ボタンでパラメータ確認のダイアログが開かれる．この機能はなんかのミスによって間違ったファイルをアップロードした時，それを確認するための機能である．この画面で下にある「実行」ボタンを押すことで，UPC にジョブを転送し，計算を開始することができる．

4.2.4　ステート画面

ステート画面を，**図 13** に示す．本画面では，実行メニューで実行することで，UPC で実行している計算プロジェクトの一覧を表示する．画面の 1 行目のプロジェクト情報は実行画面と同じで，2 行目から各パラメータファイルの情報（フォルダ名，コメント，開始時刻，終了時刻）が表示される．ここで，終了時刻が 0 になっている場合は，UPC のワーカでの計算が終わってないことを意味する．ス

図 13 ステート画面
Fig. 13 State screen.

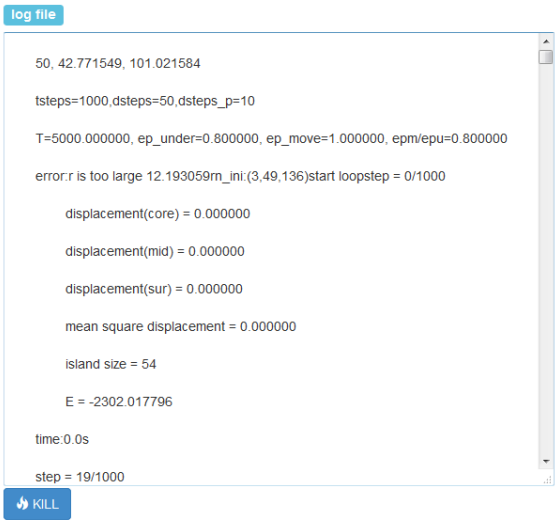

図 14 ログファイル閲覧ダイアログ
Fig. 14 Log file browsing dialog.

テート画面の最下段の「ダウンロード」をクリックすることで，ジョブの計算結果をダウンロードできる．計算結果のダウンロードは，ジョブ別にフォルダを生成し，すべての計算結果のファイルを zip 形式に圧縮して行なわれる．

Web サーバでは，各ジョブ（1 つのパラメータファイルに対応）の計算進行状況を確認できるように，UPC サーバからその計算状況のログファイルを受け取っている．ログファイルは，各パラメータの右端に表示されている「check」ボタンをクリックことで閲覧することができる．ジョブの計算が開始されていない場合にクリックすると，「計算が始まっていません」と表示したダイアログが開かれる．また，計算が進行中であれば，**図 14** に示すダイアログが開かれる．このダイアログの最下段の「kill」ボタンは，ログファイルでジョブの計算でエラーなどが発生している場合や，そのジョブの計算が不要となった場合に，そのジョブを中止させるためのものである．

5. まとめ

本研究では，本グループで実装しているユーザ PC コンピューティングシステム UPC の利便性を高めるために，Web によるユーザインターフェース機能を実現した．その際，本 Web サーバと既存の UPC サーバ間を，ファイル共有ソフトを用いて連携させることで，異なる複数の UPC サーバへのインターフェース機能の提供を可能とした．今後，本システムを様々な研究グループに提供することで，その使い勝手の評価と改善を進める予定である．

参考文献

[1] GridCafe, http://www.gridcafe.org.
[2] Globus, https://www.globus.org.
[3] 青柳有輝, 舩曳信生, 福士将, “ユーザ PC コンピューティングシステムの基本性能の実装と評価” アシュアランスシステム研究会, pp.23-28, Nov. 2014.
[4] Bittorrent, http://help.bittorrent.com/customer/portal/topics/64362-about-the-bittorrent-protocol/articles.
[5] Cohen B. “Incentives build robustness in BitTorrent,” Workshop. Economics. Peer-to-Peer Systems, vol.6, 2003.
[6] BitTorrent Sync, http://www.getsync.com.
[7] MDN Ajax, https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started.
[8] JSON, http://www.json.org.
[9] Bootstrap, http://getbootstrap.com.
[10] Apache Commons FileUpload, http://commons.apache.org/proper/commons-fileupload.